Использование непрерывного нагрузочного тестирования для оценки ёмкости и планирования ресурсов высоконагруженного сервиса

Максим Куприянов





О себе

Максим Куприянов

Яндекс Маркет

SRE

Планирую вычислительные ресурсы



Типичный web-сервис

Frontend #1
Frontend #2
Backend #1
Backend #2
Backend #3
Backend #4
Backend #5
Backend #6
Backend #7

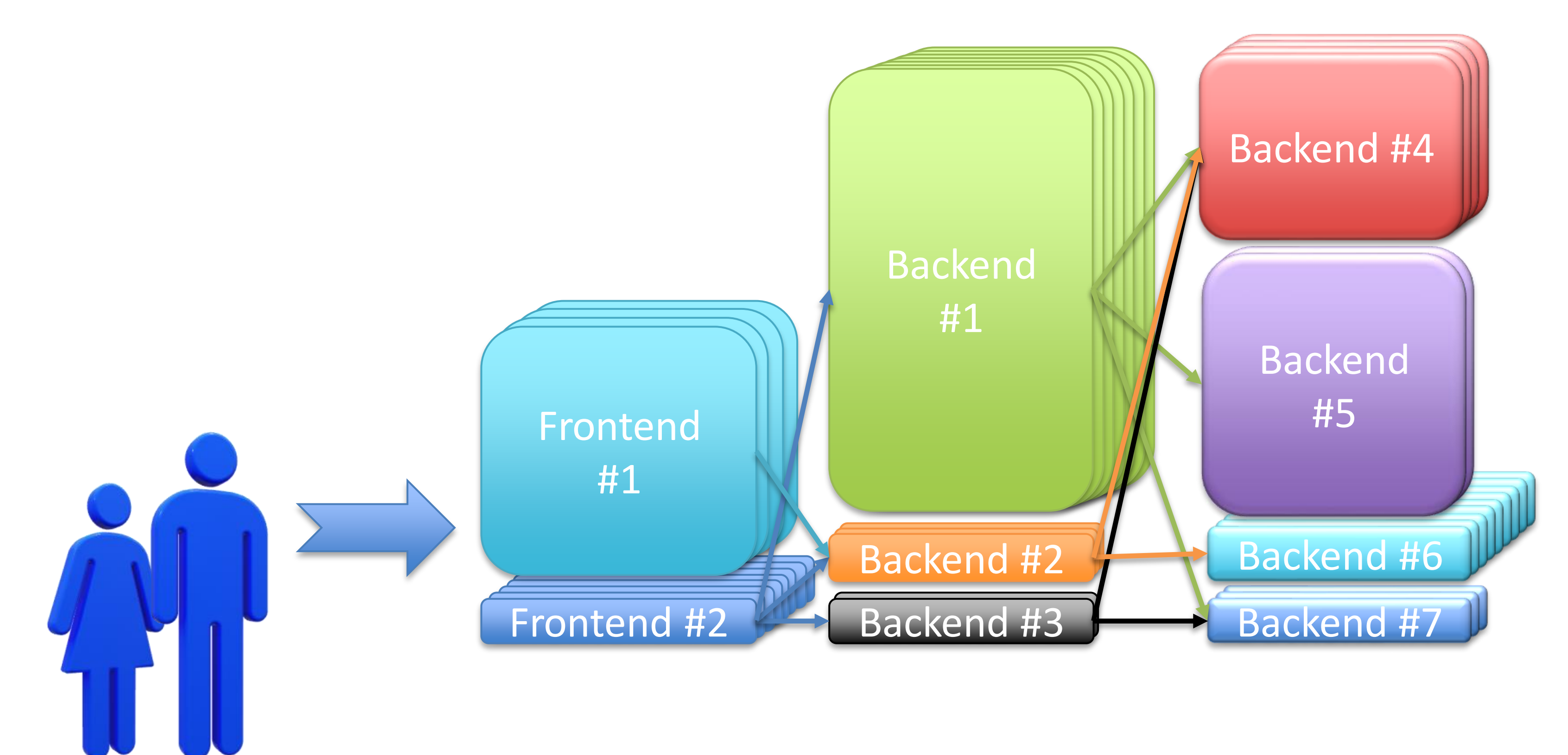


Типичный web-сервис

Frontend #1
Frontend #2
Backend #1
Backend #2
Backend #3
Backend #4
Backend #5
Backend #6
Backend #7



Типичный web-сервис

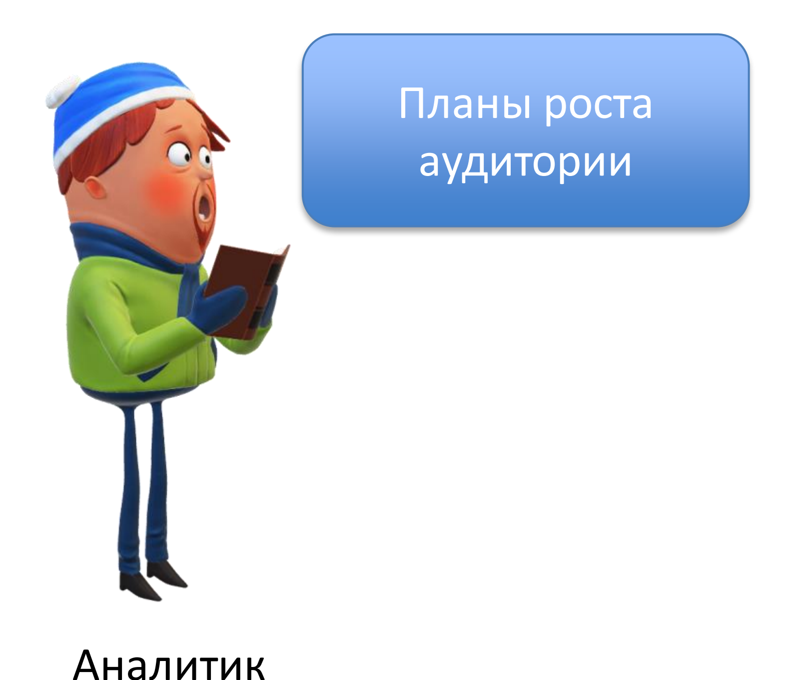


HL 2019 HighLoad++

Что нужно для прогноза?



Что нужно для прогноза?



Аналитик



Что нужно для прогноза?



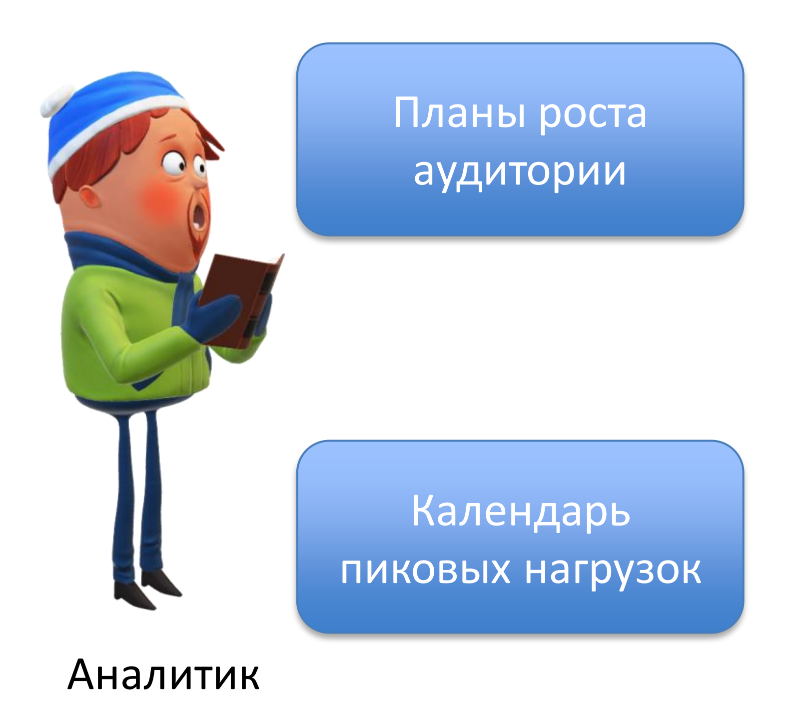
Планы роста аудитории

Календарь пиковых нагрузок

Аналитик



Что нужно для прогноза?



Планы роста аудитории

Календарь пиковых нагрузок

Аналитик



Границы масштабирования

IT Guy



Что нужно для прогноза?



Планы роста аудитории

Календарь пиковых нагрузок

Аналитик



Границы масштабирования

Текущая ёмкость сервиса

IT Guy



Что нужно для прогноза?

Планы роста аудитории

Границы масштабирования

Прогноз потребности в ресурсах

Календарь пиковых нагрузок

Текущая ёмкость сервиса

Аналитик

IT Guy



Допущения



Допущения

1. **Наша архитектура – огонь!**



Допущения

1. Наша архитектура – огонь!
2. **Поведение пользователей не хаотично**

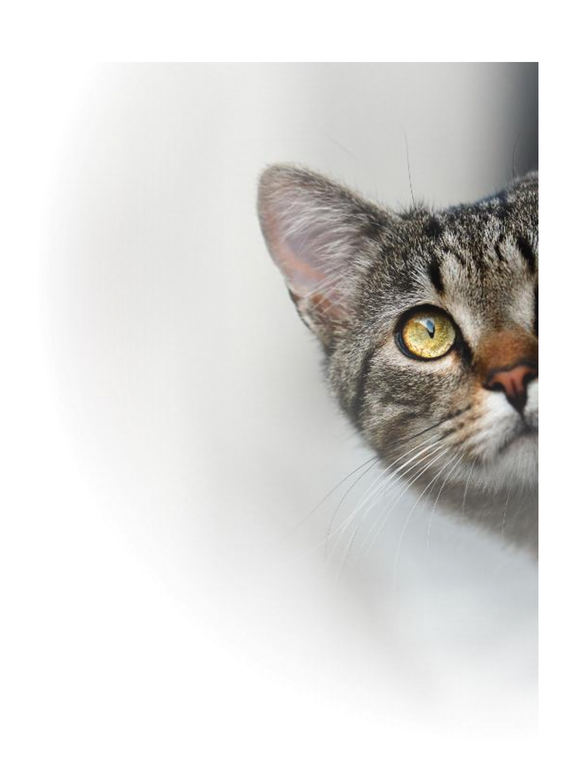


Допущения

1. Наша архитектура – огонь!
2. Поведение пользователей не хаотично
3. **20% компонентов потребляют 80% ресурсов**



Как искать ёмкость?

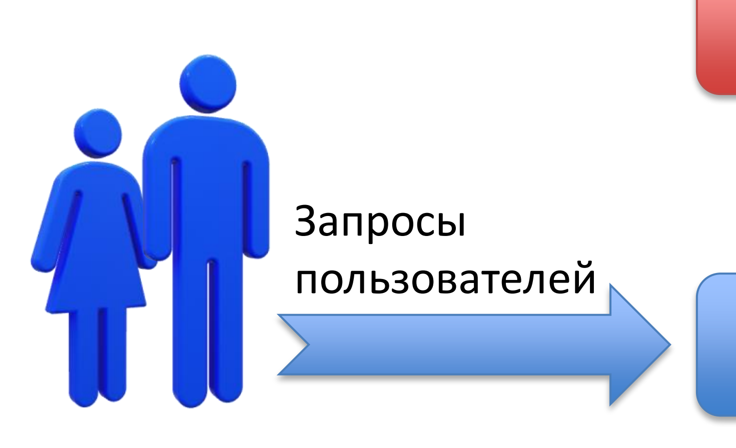




Тестирование перед релизом

Разработка

Нагрузочное тестирование



Запросы пользователей

Production-среда



Журналы сервисов



Тестирование перед релизом

Разработка

Нагрузочное тестирование

Ёмкость сервиса

Запросы пользователей

Production-среда

Журналы сервисов



Тестирование перед релизом

Разработка

Нагрузочное тестирование

Production-среда

Ёмкость сервиса

Журналы сервисов

Запросы пользователей



Тестирование перед релизом

- ✓ Быстро
- ✓ Безопасно
- ✓ Показательно при правильной организации



Тестирование перед релизом

- ✓ Быстро
- ✓ Безопасно
- ✓ Показательно при правильной организации
- ✗ Сложно правильно генерировать нагрузку
- ✗ Нужны ресурсы под стенды



Тестирование перед релизом

Q: Так себе решение, хотим что-то получше



Тестирование перед релизом

Q: Так себе решение, хотим что-то получше

A: Давайте тестировать непосредственно в production



Что нам понадобится?



Что нам понадобится?

- **Управление трафиком**





Что нам понадобится?

- Управление трафиком
- **Service Level Objectives (SLO)**





Что нам понадобится?

- Управление трафиком
- Service Level Objectives (SLO)
- **Сенсоры над инстансами**





Что нам понадобится?

- Управление трафиком
- Service Level Objectives (SLO)
- Сенсоры над инстансами
- **Работающие алерты**





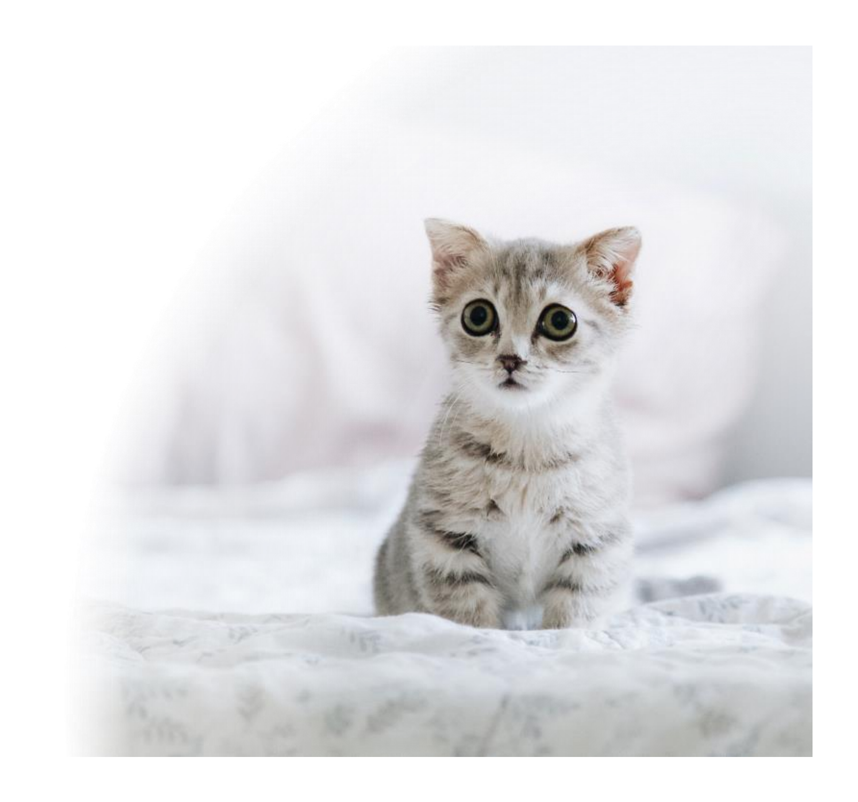
Что нам понадобится?

- Управление трафиком
- Service Level Objectives (SLO)
- Сенсоры над инстансами
- Работающие алерты
- **Надёжный выключатель**





А как грузить-то?





Нагрузка синтетикой

Владелец сервиса

Ёмкость сервиса

Генератор нагрузки

Синтезированные запросы

Журналы сервисов

Запросы пользователей

Production-среда



Нагрузка синтетикой

- ✓ Минимум дополнительных ресурсов
- ✓ Любая нагрузка
- ✓ Комплексно



Нагрузка синтетикой

- ✓ Минимум дополнительных ресурсов
- ✓ Любая нагрузка
- ✓ Комплексно
- ✗ Мусор в журналах
- ✗ Проблемы на интеграционных стыках



Нагрузка синтетикой

Q: Как избавиться от минусов решения?



Нагрузка синтетикой

Q: Как избавиться от минусов решения?

A: Будем прогружать пользовательским трафиком



Пользовательская нагрузка

Запросы пользователей

Площадка №1

Площадка №2

Площадка №3

34%

33%

33%

Балансировка



Пользовательская нагрузка

Запросы пользователей

Площадка №1

Площадка №2

Площадка №3

50%

25%

25%

Балансировка



Пользовательская нагрузка

Площадка №1
Площадка №2
Площадка №3
Запросы
пользователей
80%
10%
10%
Балансировка



Пользовательская нагрузка

Владелец
сервиса

Ёмкость
сервиса

Площадка №1

Площадка №2

Площадка №3

Запросы
пользователей

80%

10%

10%

Балансировка



Пользовательская нагрузка

- ✓ Минимум дополнительных ресурсов
- ✓ Комплексно



Пользовательская нагрузка

- ✓ Минимум дополнительных ресурсов
- ✓ Комплексно
- ✗ Трафик ограничен
- ✗ Структура трафика меняется



Пользовательская нагрузка

Q: Что делать с дефицитом пользовательской нагрузки?



Пользовательская нагрузка

Q: Что делать с дефицитом пользовательской нагрузки?

A: Будем тестировать по одному инстансу от компонента. Ёмкость будем измерять в "запросах в секунду" (RPS).



Пользовательская нагрузка

Q: Что делать с изменением структуры трафика во времени?



Пользовательская нагрузка

Q: Что делать с изменением структуры трафика во времени?

A: Будем тестировать непрерывно (24x7)!



Непрерывное тестирование

Запросы
пользователей
(M RPS)

Инстанс №1

Инстанс №2

Инстанс №Z

Балансер



Непрерывное тестирование

Запросы
пользователей
(M RPS)

Инстанс №1

Инстанс №2

Инстанс №Z

N RPS

M-N RPS

Сплиттер

Балансер



Непрерывное тестирование

Владелец сервиса

Ёмкость компонента, RPS =
«Ёмкость инстанса №1» * Z

Инстанс №1

Инстанс №2

Инстанс №Z

Запросы пользователей (M RPS)

N RPS

M-N RPS

Сплиттер

Балансер



Непрерывное тестирование

Текущая нагрузка

Ёмкость



Непрерывное тестирование

Текущая нагрузка

Утилизация компонента

Ёмкость



Непрерывное тестирование

Утилизация компонента

Ёмкость



Непрерывное тестирование

Утилизация компонента

Прогноз потребности в ресурсах

Ёмкость



Непрерывное тестирование

Детали реализации





Непрерывное тестирование

Балансер

Инстанс №2

Инстанс №3

Инстанс №Z

M-N
RPS

Fallback

Инстанс №1

N RPS

N RPS

Сплиттер



Непрерывное тестирование

Балансер
Инстанс №2
Инстанс №3
Инстанс №Z
N RPS
M-N RPS
Fallback
Инстанс №1
0 RPS
N RPS
Сплиттер



Непрерывное тестирование

Эталонные инстансы

Балансер

Инстанс №2

Инстанс №3

Инстанс №Z

Эталонные метрики

0 RPS

M-N RPS

Fallback

N RPS

Инстанс №1

Пишем метрики

База с метриками

N RPS

Сплиттер



Что входит в метрики?



Что входит в метрики?

- RPS



Что входит в метрики?

- RPS
- Квантили таймингов



Что входит в метрики?

- RPS
- Квантили таймингов
- Ошибки в различных разрезах



Непрерывное тестирование

Эталонные инстансы

Балансер

Инстанс №2

Инстанс №3

Инстанс №Z

Эталонные метрики

0 RPS

M-N RPS

Fallback

N RPS

Инстанс №1

Пишем метрики

База с метриками

N RPS

Сплиттер



Непрерывное тестирование

Балансер

Инстанс №2

Инстанс №3

Инстанс №Z

0 RPS

M-N
RPS

Fallback

Инстанс №1

N RPS

База с
метриками

N RPS

Читаем метрики

Сплиттер

Управляющий
контур



Непрерывное тестирование

Балансер
Инстанс №2
Инстанс №3
Инстанс №Z
0 RPS
M-N
RPS
Fallback
N RPS
Инстанс №1
База с
метриками
N RPS
Сохраняем значение N
Меняем значение N
Сплиттер
Управляющий
контур



Непрерывное тестирование



Управляющий контур

Логика работы



Управляющий контур

- **Собираем достаточный набор метрик**



Управляющий контур

- Собираем достаточный набор метрик
- **Если инстанс под нагрузкой не показывает деградацию – повышаем N**



Управляющий контур

- Собираем достаточный набор метрик
- Если инстанс под нагрузкой не показывает деградацию – повышаем N
- **Если показывает деградацию – понижаем N**



Управляющий контур

- Собираем достаточный набор метрик
- Если инстанс под нагрузкой не показывает деградацию – повышаем N
- Если показывает деградацию – понижаем N
- **Если есть сомнения – не делаем ничего**



Живые примеры



Тестируемый и эталонные инстансы

Трафик на инстанс

RPS

800
600
400
200
0

08:00 10:00 12:00 14:00 16:00 18:00 20:00 22:00 00:00

Под нагрузкой — Эталон 1 — Эталон 2



Ёмкость компонента на графиках

Утилизация текущей ёмкости, %

120.0%
100.0%
80.0%
60.0%
40.0%
20.0%

9/8 9/16 9/23 10/1 10/8 10/16 10/24 11/1 11/8 11/16 11/23 12/1

Все локации — Без 1 ДЦ



Непрерывное тестирование

Q: Что можно еще улучшить?



Непрерывное тестирование

Q: Что можно еще улучшить?

A: Есть ресурсы под отдельную площадку для компонента? Поднимем там «теневой инстанс». Копируем на него трафик. Ответы игнорируем.



Копируем трафик

Инстанс №1

Инстанс №Z

Запросы пользователей

M RPS

Балансер



Копируем трафик

Ёмкость компонента, RPS =
«Ёмкость теневого инстанса» * Z

Владелец сервиса

Теневой инстанс

Инстанс №1

Инстанс №Z

Запросы пользователей

M RPS

Сплиттер

Копия N RPS

M RPS

Балансер



Когда методика не работает



Когда методика не работает

- На сервисах, для которых тайминги – не главное



Когда методика не работает

- На сервисах, для которых тайминги – не главное
- Когда тайминги настолько критичны, что любое их колебание недопустимо (тут поможет теневой инстанс)



Когда методика не работает

- На сервисах, для которых тайминги – не главное
- Когда тайминги настолько критичны, что любое их колебание недопустимо (тут поможет теневой инстанс)
- Когда трафика на локации недостаточно для того, чтобы загрузить несколько инстансов



Советы



Service Level Objective

Как с ним работать?



Service Level Objective

- Несколько разных SLO на компонент. Например, типовые запросы или разные квантили



Service Level Objective

- Несколько разных SLO на компонент. Например, типовые запросы или разные квантили
- SLO для capacity тестирования жёстче, чем SLA для пользователя на 10% и более



Service Level Objective

- Несколько разных SLO на компонент. Например, типовые запросы или разные квантили
- SLO для capacity тестирования жёстче, чем SLA для пользователя на 10% и более
- Регулярно уточняйте SLO



Управляющий контур

- **Собираем достаточный набор метрик**
- Если инстанс под нагрузкой не показывает деградацию – повышаем N
- Если показывает деградацию – понижаем N
- Если есть сомнения – не делаем ничего



Достаточный набор метрик

Как собираем?



Достаточный набор метрик

- Собираем на интервалах не уже, чем стандартный таймаут ответа сервиса, например, 1 раз в минуту



Достаточный набор метрик

- Собираем на интервалах не уже, чем стандартный таймаут ответа сервиса, например, 1 раз в минуту
- Необходимо несколько интервалов, например, 5



Достаточный набор метрик

- Собираем на интервалах не уже, чем стандартный таймаут ответа сервиса, например, 1 раз в минуту
- Необходимо несколько интервалов, например, 5
- Отсекайте шум медианой



Управляющий контур

- Собираем релевантный набор метрик
- Если инстанс под нагрузкой не показывает деградацию – повышаем N
- Если показывает деградацию – понижаем N
- **Если есть сомнения – не делаем ничего**



Когда сомневаться?



Когда сомневаться?

- Мало трафика



Когда сомневаться?

- Мало трафика
- Деградация эталонных инстансов



Когда сомневаться?

- Мало трафика
- Деградация эталонных инстансов
- Ночь. Вам точно нужны ночные результаты?



Безопасный RPS

Что это?



Безопасный RPS

- Зафиксируйте и регулярно синхронизируйте с владельцами компонента



Безопасный RPS

- Зафиксируйте и регулярно синхронизируйте с владельцами компонента
- Если управляющий контур **долго** не может принять решение – уходите на безопасный RPS



Безопасный RPS

- Зафиксируйте и регулярно синхронизируйте с владельцами компонента
- Если управляющий контур **долго** не может принять решение – уходите на безопасный RPS
- **«Долго»** – индивидуально для каждого компонента



Максимальный RPS

- Зафиксируйте для каждого тестируемого компонента RPS, выше которого не нужно пытаться увеличить число **N**



Журналы/логи



Журналы/логи

- На нагруженном инстансе логи съедят всё место



Журналы/логи

- На нагруженном инстансе логи съедят всё место
- Пишите их буферизовано и лучше сразу в сеть



Сеть

- Типовой SLA по доступности даже для локальной сети – 99%. Вероятнее всего, вам не нужен SLO выше, чем 99 перцентиль.



Как разбираться с деградациями



Как разбираться с деградациями

- Завести таймлайн с релизами различных компонентов и включением экспериментов



Как разбираться с деградациями

- Завести таймлайн с релизами различных компонентов и включением экспериментов
- Очень помогает трассировка по логам



Всем точных прогнозов!

Веня

Максим Куприянов

SRE

Яндекс Маркет

maxk@yandex-team.ru

